REMARKS

Claims 1-32 are pending in the present application.

Claims 1-32 stand rejected under 35 U.S.C. §102(b) as being anticipated by Liencres et al. (U.S. Patent No. 5,434,993) (hereinafter "Liencres"). Applicant respectfully traverses this rejection.

Applicant's claim 1 recites

"A system, comprising:

a node including an active device, a memory, and an interface coupled by an address network and a data network;

an additional node coupled to send a coherency message to the interface in the node via an inter-node network, wherein the coherency message requests an access right to a coherency unit;

wherein in response to the coherency message, the interface is configured to send **a first type of address packet on the address network if** a global access **state** of the coherency unit in the node **is a modified state** and to send a **second type of address packet if** the global access **state** is **not the modified** state;

wherein in response to the second type of packet, the memory is configured to send a data packet corresponding to the coherency unit on the data network, regardless of whether the memory has an ownership responsibility for the coherency unit."

The Examiner asserts Liencres teaches each and every limitation recited in Applicant's claim 1. More particularly, the Examiner asserts Liencres teaches in FIG. 3a, the interface to an internode network as being element 31, an address network coupling the active device the interface and the memory as being element 33. In addition, the Examiner asserts Liencres teaches the remaining limitations at col. 8, line 63 through col.

9, line 8, and at col. 2 lines 22-30. Applicant respectfully disagrees with the Examiner's characterization of Liencres and the application of Liencres to Applicant's claims.

Specifically, as illustrated in Fig.3a and 3b of Liencres, element 33 cannot be the address network as recited in Applicant's claim 1 because it only couples the bus controller 31 to the processor cache controller 35. It does not couple the active device, the interface, and the memory as recited in claim 1.

However, the above notwithstanding, Applicant further asserts Liencres actually discloses at col. 8 line 63 - col. 9, line 8

> "1. If another processor subsystem issues a write transaction to an address corresponding to an owned subblock before the subblock is written back to main memory by the pending write-back controller 40, the write-back must not occur since the pending write-back controller 40 contains "stale" data.
>
> 2. If another processor subsystem issues a read request packet to an address corresponding to a subblock owned by the pending write-back controller 40 before the subblock is written back to main memory, or before another processor issues a write transaction to an address corresponding to the same subblock, then the pending write-back controller 40 must reply with a read reply packet." (Emphasis added)

From the foregoing, Applicant submits Liencres is discussing write-back protocol. Applicant asserts Liencres is disclosing not performing a write-back if another processor issues a write to that data block, and that a reply including the requested data is supplied by the owning processor even if the data is modified. Applicant fails to see how this teaches sending one type of address packet if the data is modified and another type of address packet if the data is not modified.

Liencres also discloses at col. 2, lines 16-59

> "Processors 2 and 3 perform load A operations to obtain the value of A. During each processors load operation, the value of A is stored in the processor's local cache memory. Processors 2 and 3 now "share" memory location A and both caches have "valid" data. In FIG. 1b, Processor 1 has written a value of 2 to location A. This is permitted since neither processor 2 or processor 3 "owned" memory location A. In order to change the

> contents of memory location A, Processor 1 broadcasts a message across the memory bus informing other memory devices that the contents of memory location A has changed. This message causes the cache memories of processor 2 and 3 to change the status of memory location A to "invalid". The main memory unit does not maintain a set of status bits for each memory line. Instead, the main memory monitors a control line on the memory bus that is asserted whenever a request is made for a memory line that is "owned" by a processor subsystem. When the "owned" control line is asserted, the main memory learns that the line is owned by some processor subsystem and therefore does not respond to the request. Cache memory 1 now "owns" location A since it modified the contents of memory location A without updating the main memory. In FIG. 1c, processor 1 has changed the contents of memory location A to the value of 3. Since processor 1 does not share memory location A with any other processor, Processor 1 does not need to send a message across the memory bus. However, in FIG. 1d, processor 3 requires the value of memory location A for a load operation. Processor 3 must therefore send a request across the bus requesting the value of memory location A. Since processor 1 "owns" memory location A, it must respond to the request with a reply containing contents of memory location A. Memory location A is now represented in the cache memories of processors 1 and 3. Although memory location A is still "owned" by processor 1, it must now "share" memory location A with processor 3. Therefore, any further changes to memory location A by processor 1 must be forwarded to processor 3. Processor 1 must eventually write-back the changed contents of memory location A to main memory."

From the foregoing, Liencres is discussing how ownership changes, and how the system determines if a line is owned by monitoring a control line on a memory bus. Liencres further discloses an owning processor (1) sending the requested data to the requesting processor (3). Applicant asserts this is NOT responding **to a specific one of two kinds of packets** (one that indicates the data is not modified) by sending data regardless of whether a coherency unit is owned or not.

Thus, although Liencres discloses a mechanism for maintaining coherency, Liencres does so in a different way.

Accordingly, Applicant submits Liencres does not teach or disclose "an interface coupled by an address network and a data network," and "in response to the coherency message, the interface is configured to send **a first type of address packet on the**

address network if a global access **state** of the coherency unit in the node **is a modified state** and to send a s**econd type of address packet if** the global access **state** is **not the modified** state;" and "wherein in response to the second type of packet, the memory is configured to send a data packet corresponding to the coherency unit on the data network, regardless of whether the memory has an ownership responsibility for the coherency unit" as recited in claim 1.

Accordingly, Applicant submits claim 1, along with its dependent claims, patentably distinguishes over Liencres for the reasons given above.

Applicant's claims 13 and 24 recite features that are similar to the features recited in claim 1. Thus Applicant submits claims 13 and 24, along with their respective dependent claims, patentably distinguish over Liencres for at least the reasons given above.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-94901/SJC.

Respectfully submitted,

/ Stephen J. Curran /

Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: May 1, 2007